

REMARKS

In the final Office Action dated January 3, 2007, Claims 1 and 3-17 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Microsoft Corporation's *Microsoft Windows Management Instrumentation Scripting*, April 1999, pp. 1-15 (hereinafter MSWMI). Claims 8 and 17 were also rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter.

With this response, Claims 1 and 3-17 remain pending.

For the reasons set forth below, applicants respectfully request reconsideration and allowance of the pending claims. However, prior to discussing those reasons, a brief description of aspects of the claimed subject matter and of the cited reference, MSWMI, is presented. These descriptions are presented to assist the Examiner in appreciating the differences between the claimed subject matter and the cited reference, and should not be viewed as limiting on the disclosed subject matter.

Description of the Claimed Subject Matter

The disclosed subject matter is directed to resolving the issue of accessing objects (software and hardware) for the purpose of obtaining and/or setting information at those objects, which objects are external to a managed code environment.

As those skilled in the art will readily appreciate, a managed code environment creates a virtual "world" in which applications compiled to an intermediate form can execute. Advantageously, applications compiled to this intermediate form are executable on any processor and operating system, so long as that processor/operating system hosts the managed code environment. It's a "write once, play anywhere" application. For these reasons, it is said of such applications that they are platform independent or platform agnostic. Of course, one example of

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206.682.8100

a managed code environment, as disclosed in the application, is Microsoft Corporation's .NET platform.

One of the drawbacks to a managed code environment has been that, to an application executing within the managed code environment, the boundaries of the managed code environment define the boundaries of the accessible "world." Unfortunately, critical objects, devices, processes, etc., while operating on the same computer as the managed code environment, operate and are accessible only from outside of the managed code "world." This managed code "boundary" has limited the range of features that a managed code application can address. The novel subject matter directly addresses this shortcoming of managed code environments.

The claimed subject matter includes providing a platform-independent API within a managed code environment that a managed code application can call to access external objects. Since the API is platform-independent, the call will include the platform-specific information. A typical request from an application to a generic API might include information such as the identification of a specific object external to the managed code environment, details for accessing that object, how to locate the object, and the like. The request is forwarded to the identified object according to the information in the request. Correspondingly, a response is received to the call/request, the content of the response is formatted in a manner compatible with the managed code environment, and the formatted response is returned to the requesting managed code application.

Description of MSWMI

MSWMI describes an overview of Windows Management Instrumentation (WMI) for Microsoft Windows platforms, and in particular, this describes the WMI scripting API that

allows various scripting languages to access the functionality implemented by WMI on Microsoft Windows computers.

As MSWMI describes, and as those skilled in the art will readily recognize, WMI is Microsoft's solution for providing access to management type data on its Windows platforms, including enterprise environments. Using the WMI scripting tools, applications (including script based applications) can be developed to reduce the complexity and cost of enterprise management. Enterprise environments or networks, as can be readily discovered in all computer-related dictionaries, refers to networks (or interconnected networks) of computer systems owned by the enterprise, configured according to the enterprise's needs including diverse geographic locations and encompassing a range of platforms, operating systems, protocols, and network architectures. However, what an enterprise environment is not is a managed code environment to which all applications that run therein are compiled to an intermediate form, and which managed code environment defines virtual boundaries of the applications. Indeed, while the various enterprise environments include many platforms, each platform could host a managed code environment. However, the enterprise environment is a description of a collection of various computers, certainly not a reference to a managed code environment.

MSWMI references various scripting languages, including VBScript and Jscript. These are languages that require an interpreter, or scripting engine, to execute. However, applicants submit that those skilled in the art will readily appreciate the distinctions between a scripting engine and a managed code environment.

35 U.S.C. § 101 Rejections

Claims 8 and 17 were rejected under 35 U.S.C. § 101 as being directed to unpatentable subject matter, and particularly, that no mention is made as to the hardware on which the methods execute.

Applicants have amended Claim 8 as follows:

A computer-controlled apparatus **comprising a processing unit and a system memory, and wherein the apparatus further comprises a managed code runtime environment** and is configured to carry out the method of any one of Claims 1 and 3-6. (Emphasis added).

Claim 17 is similarly amended.

Applicants submit that these claims now recite patentable subject matter, and request that the 35 U.S.C. § 101 be withdrawn, and the claim allowed.

35 U.S.C. § 103(a) Rejections

The Office Action rejected Claims 1 and 3-17 as being obvious in view of MSWMI and acknowledged prior art. Applicants respectfully traverse the rejections.

Claim 1

The Office Action cites to MSWMI and its discussion of instrumentation data available from WMI, as well as to the background of the application (referred to as admitted prior art or "APA"), and suggests that together they teach or suggest each element of Claim 1, and further that it would have been obvious to one of ordinary skill in the art to combine the teachings of the two. More particularly, the Office Action apparently suggests that since WMI is a Microsoft product, and since the .NET platform is a Microsoft product, it would have been obvious to one of ordinary skill in the art to incorporate a WMI interface into the .NET platform. Applicants disagree.

Applicants assert, and the Office Action also admits, that applications written for execution in a managed code environment are "platform independent designed to communicate with many other sources." Office Action, page 6. As platform independence is a goal of a managed code environment, it would be quite contradictory to incorporate platform specific technology into the environment. Indeed, at least one significant issue identified in the

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{LLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206 682 8100

background of the application (the APA) regarding managed code environments is that managed code environments are, by their nature, independent of platform specific functions/data, and that access to those platform specific functions/data from within a managed code world has been unavailable. In this light, applicants assert that it would not be obvious, but contradictory, to develop a platform independent environment and simply incorporate platform specific API interfaces, as the Office Action suggests. Accordingly, applicants submit that one skilled in the art would not be motivated to make such combinations.

The Office Action's assertion that, as both WMI and .NET (a managed code environment) are Microsoft products, it would have been obvious to combine the teachings is contrary to established rules for determining obviousness. This assertion, if maintained, suggests that some inventors, such as those from a particular company, can be held to higher standards with regard to what is and is not obvious, than others outside of that company. Indeed, the Office Action's assertion turns 35 U.S.C. § 103(a) upside-down, suggesting that obviousness is defined as a function of where you work, and not as a function of what is obvious to a hypothetical person of ordinary skill in the art.

Finally, applicants further submit that the Office Action's assertions are a product of impermissible hindsight reasoning. Indeed, the background of the application (the APA) describes a desire to access platform specific instrumentation data, such as information via WMI calls, from within a managed code environment, such as a .NET platform. From this perceived need grew the present invention. However, it appears to applicants as though the Office Action has essentially relied on the problem and solution proposed in the application, reciting them as the motivating factors to one of ordinary skill to make the combination of the WMI and .NET platforms. Applicants submit that this is the definition of impermissible hindsight reasoning.

It is well established that to reach a proper, *prima facie* conclusion of obviousness, the Examiner must look back in time from when the invention was unknown and just before it was

made. Moreover, the evaluation must be from the viewpoint of a hypothetical person of ordinary skill in the art. However, while the tendency to resort to "hindsight" analysis and reasoning based upon applicants' disclosure is often difficult to avoid due to the very nature of the examination process, such hindsight reasoning must be avoided and any obviousness conclusion must be reached on the basis of the facts gleaned from the prior art. M.P.E.P. § 2142. This, of course, means that the teaching or suggestion to make the claimed combination must be found in the prior art, and not based on applicants' disclosure. *In re Vaack*, 947 F.2d 488, 20 U.S.P.Q.2d 1438 (Fed. Cir. 1991).

Applicants submit that the combination suggested by the Office Action, i.e., combining a WMI interface into a platform independent managed code environment is contradictory to the nature of the independent of the environment, and one would not be motivated to make this modification. Applicants also submit that an improper standard (based on the company for whom the inventor worked and not based on the hypothetical person of ordinary skill in the art) was used in determining whether the elements of Claim 1 are obvious in view of MSWMI. Moreover, applicants submit that the Office Action's conclusion of obviousness was based on the applicants' disclosure and involved improper hindsight reasoning. In light of these assertions, applicants submit that a proper *prima facie* case of obviousness is not made. Applicants therefore request that the 35 U.S.C. § 103(a) rejection of Claim 1 be withdrawn, and the claim allowed.

Claims 3-8

Claims 3-8 each depend, directly or indirectly, from independent Claim 1. When read in combination with independent Claim 1, applicants submit that Claims 3-8 are also in condition for allowance and request that the 35 U.S.C. § 103(a) rejections be withdrawn, and the claims allowed.

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206 682 8100

Claim 9

Independent Claim 9 was rejected for similar rationale as set forth in regard to Claim 1. Accordingly, for the same reasons as discussed above in regard to Claim 1, applicants submit that Claim 9 is also in condition for allowance, and request that the 35 U.S.C. § 103(a) rejection be withdrawn, and the claim allowed.

Claims 11-17

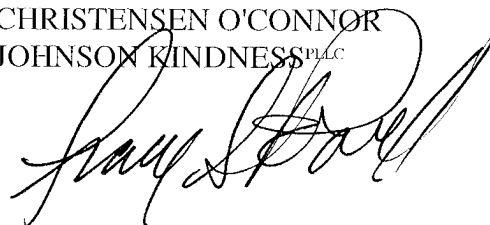
Claims 11-17 each depend, directly or indirectly, from independent Claim 9. When read in combination with independent Claim 1, applicants submit that Claims 11-17 are also in condition for allowance and request that the 35 U.S.C. § 103(a) rejections be withdrawn, and the claims allowed.

CONCLUSION

Applicants submit that the pending claims are now in condition for allowance. Reconsideration and early allowance of the pending claims is requested. If the Examiner has any questions regarding this matter, the Examiner is invited to contact the applicants' representative at the number below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^{PLLC}



Tracy S. Powell
Registration No. 53,479
Direct Dial No. 206.695.1786

TSP:lal

LAW OFFICES OF
CHRISTENSEN O'CONNOR JOHNSON KINDNESS^{PLLC}
1420 Fifth Avenue
Suite 2800
Seattle, Washington 98101
206 682 8100